Computational Design of Stable Planar-Rod Structures

Eder Miguel Mathias Lepoutre Bernd Bickel

IST Austria

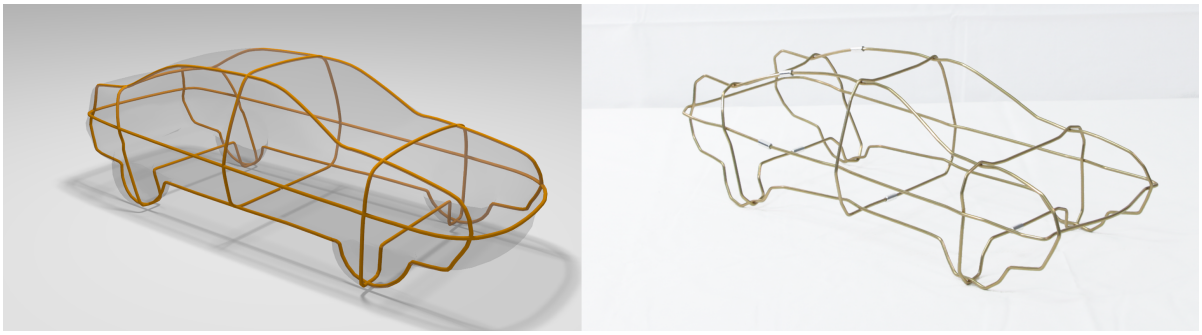

Figure 1: *Given a set of planar input contours (left), our system computes a stable, self-supporting wire sculpture. The physical prototype (right) can be easily fabricated using a 2D wire bending machine and assembled without the need of connectors between crossing wires.*

Abstract

We present a computational method for designing wire sculptures consisting of interlocking wires. Our method allows the computation of aesthetically pleasing structures that are structurally stable, efficiently fabricatable with a 2D wire bending machine, and assemblable without the need of additional connectors. Starting from a set of planar contours provided by the user, our method automatically tests for the feasibility of a design, determines a discrete ordering of wires at intersection points, and optimizes for the rest shape of the individual wires to maximize structural stability under frictional contact. In addition to their application to art, wire sculptures present an extremely efficient and fast alternative for low-fidelity rapid prototyping because manufacturing time and required material linearly scales with the physical size of objects. We demonstrate the effectiveness of our approach on a varied set of examples, all of which we fabricated.

Keywords: computational fabrication, physics-based modeling, optimization, illustrative fabrication

Concepts: •**Computing methodologies** → **Shape modeling;** *Physical simulation;* •**Mathematics of computing** → *Geometric topology;* •**Applied computing** → Computer-aided design;

1 Introduction

Wires are an extraordinary sculptural material. Lightweight and at the same time strong, they are used as the basis for impressive wire sculptures, allowing artists to abstract and visually approximate shapes in three dimensions. Artifacts such as the wire sculptures by Alexander Calder [1] and Jakobi [2] demonstrate their expressiveness and indicate their power as a means to represent shapes (see Fig. 3). While wire sculptures usually emerge from a manual, artistic process, their potential in bringing virtual models to the real world is largely unexplored.

Recently, the maker community has introduced computer-controlled wire bending machines that allow the generation of wires following complex paths. However, it remains unclear how to convert a given 3D surface model into a fabricable representation. This process poses several challenges: finding an optimal set of wires to cover geometric features is highly challenging and subject to aesthetic considerations. Potential wire shapes are limited by fabrication constraints, and the model itself needs to be assemblable from individual primitives. Furthermore, while some wire figures can be self-supporting, i.e., stable without soldering or other additional connectors, finding a stable configuration is extremely challenging due to numerous complex interactions of contact forces.

To address these issues, we propose a computational approach that allows artists and non-expert users to intuitively design and fabricate self-supporting wire sculptures. In this work, we deliberately focus on self-supporting structures, as they guarantee easy assemblability without the need for any additional equipment.

Our approach starts with a closed surface mesh and then allows the user to define desired wire contours resulting from plane-surface intersections and explore their effect on the stability of the figure. We provide an optimization-in-the-loop, where the system optimizes the discrete ordering of wire intersections and rest shape of the elastic wires. Our system also provides intuitive control over the inherent trade-off between staying faithful to the input shape and stability. With our tool we can quickly explore designs while taking into account the coupled problem of aesthetic shape approximation and physical constraints. The output of our system can be directly sent to a 2D wire bending machine, and after the bending process, the figure is ready to be assembled by simply following the provided assembly sequence.

We believe our technique is a valuable contribution to the rapid prototyping community. In contrast to 3D printers or any other method

SIGGRAPH '16 Technical Paper,, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925978

that either fabricates a volume or surface, wire sculptures scale linearly with the size of the object. They are also extremely fast to fabricate. The simplicity in fabrication requires the satisfaction of complex physical constraints. We develop a model for wire sculptures that respects important fabrication constraints and propose an optimization strategy that computes the arrangement of wires and their shape to improve stability. We demonstrate our complete system for wire sculpture design for several complex input models. We validate the feasibility of the resulting designs by fabricating numerous physical wire sculptures and performing drop tests.

2 Previous Work

Illustrative Fabrication In recent years, computer graphics has contributed significantly to computational design tools that employ fabrication technologies as a means to create a stylistic representation of a digital shape, rather than an exact 3D reproduction. Often, these abstractions are motivated by fabrication constraints of the underlying fabrication method, such as the milling of height fields [Weyrich et al. 2007], the laser cutting of planar elements [Hildebrand et al. 2012; Cignoni et al. 2014], or the use of a specific material or primitives, such as plastic pipes [Sageman-Furnas et al. 2015], in the process.

Starting from a planar sheet, mathematical folding algorithms [Demaine and ORourke 2007; Kilian et al. 2008] allow the creation of intricate 3D structures. Relaxing the fabrication constraints and allowing cutting, bending, and gluing of strips [Mitani and Suzuki 2004; Massarwi et al. 2007] allows the recreation of complex surface models. Recent work also investigated computational approaches for designing shapes under the assumption that the material allows shearing [Garg et al. 2014], as in wire meshes, or stretching, as in balloons [Skouras et al. 2012]. Iarussi et al. [2015] investigated the creation of 2D wire sculptures, with the goal of creating 2D wire-wrapped jewelry by segmenting a drawing into a small number of wires and bending the wires to give them shape. In contrast, we assume that our contours are provided by the user, our sculptures are 3D, and we do not allow any connectors. Inspired by this line of work, our goal is to provide a computational tool that supports the design and exploration of wire sculptures, respects the fabrication constraints inherent to the underlying material and manufacturing process, and alleviates the user's need to anticipate complex underlying physical effects during the design.

Interlocking and Self-Supporting Structures An interesting line of work involves computational design tools for creating physical artifacts consisting of parts that can be easily assembled without the need for glue, bolts, or any other connector type. One of the most prominent examples are self-supporting structures in architecture. Whiting et al. [2009; 2012] presented methods to automatically adjust architectural models to guarantee structural stability. Recent work also investigated the design of valid self-supporting shapes [Vouga et al. 2012; Liu et al. 2013; De Goes et al. 2013; Panozzo et al. 2013] and the decomposition of 3D shapes into self-supporting, discrete-element assemblies [Frick et al. 2015]. While most of this work has adapted the shapes of the individual elements, Skouras et al. [2015] presented an interactive design system for surfaces consisting of flexible interlocking quadrilateral elements of a single size and shape.


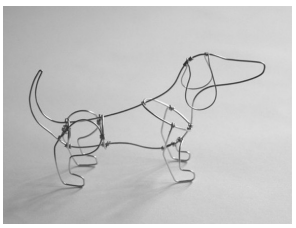

Figure 3: *Our work is inspired by wire sculptures, such as the elephant by Alexander Calder* [1] *or the dog by Jakobi* [2].

For many of these structures, finding valid assembly sequences is a non-trivial problem, and has been investigated in the context of 3D puzzles [Song et al. 2012], jigsaw puzzles [Lau et al. 2014], interlocking figures consisting of planar elements [Hildebrand et al. 2012; Cignoni et al. 2014], and furniture assembly [Fu et al. 2015]. Deuss et al. [2014] computed a work-minimizing assembly sequence for self-supporting structures that are composed of bricks or stone blocks without any mortar. Conceptually similar to our work, their solution leverages the internal force distribution and only provides the minimally required additional supports to keep the structure in static equilibrium at all stages of the assembly. However, our problem requires a significantly different solution strategy, as the nature of the underlying problem is different. In contrast to having a predetermined layout of rigid bricks, we do not know the ordering of wires at contact locations, we have elastic elements, and we are able to adapt the shape and thereby influence the frictional contact forces.

Simulation and Optimization of Rods Many researchers have developed approaches for simulating elastic rods, ranging from mass-spring [Selle et al. 2008; Iben et al. 2013] to multi-body [Hadap 2006] and Cosserat models [Pai 2002; Bertails et al. 2006; Grégoire and Schömer 2007]. In our work, we build on the discrete elastic rods model developed by Bergou et al. [2008] and represent each of our polylines as inextensible elastic rods.

The design and optimization of rod networks was recently investigated by Perez et al. [2015]. They design flexible rod meshes by adjusting the cross-sectional profiles of the rods and their rest centerline to best approximate the target deformations. This approach allows the local control of the bending and stretching resistance of the surface with a single material. While our approach shares some similarities, there are several conceptual differences that require a different solution strategy. Their rod meshes (a) are rigidly connected, (b) are fabricated by 3D printing the object as a single piece, (c) are flexible with the goal to match deformed poses under specific force loads, and (d) use the thickness of rods as degrees of freedom. By contrast, our wire sculptures (a) are stable due to contact and friction forces, (b) are manufactured by the assembly of individual curved segments, (c) are optimized with the goal of providing self-support and structural stability, and (d) exploit the discrete ordering of crossings and local curvature as degrees of freedom.

In the broader context, frictional contact is also important for modeling hair. Derouet-Jourdan [2013] treated hair shape as a static equilibrium configuration of a hair simulator and solve an inverse problem for finding a physically valid rest shape for a given hairstyle. Twigg and Kačić-Alesić [2011] proposed a solution to the inversion problem with contact of mass-springs systems to reduce sagging in animations through nonlinear optimization. In our work, we go beyond finding static equilibrium and optimize for self-supporting structures that are as stable as possible even under external load cases.

[1] Sculpture: ©2016 Calder Foundation, New York / Bildrecht, Wien; Photo: ©2016 Calder Foundation, New York / Art Resource, NY

[2] http://jakobi.etsy.com/

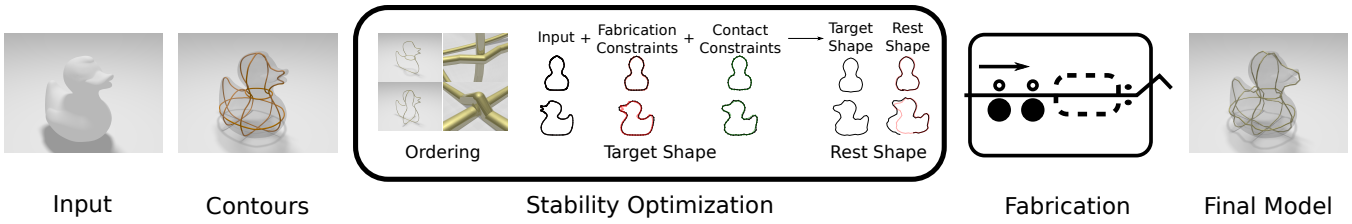

Figure 2: *Overview of our workflow. The user provides a target shape and indicates a set of desired contours by placing planes and computing their intersection with the input shape surface. Our optimization scheme automatically computes an insertion sequence, layering, and rest shape of the flexible wires such that the resulting figure closely approximates the desired contours and is stable. Physical prototypes can be fabricated using a wire bending machine.*

3 Overview

Given a 3D shape, our system allows the user to explore planar-rod-based designs. An overview is illustrated in Fig. 2. Each plane can generate several contours, from which the user selects the desired ones. Then, the system automatically tries to compute a stable wire structure that closely resembles the provided contours while being structurally stable at the same time. Finding an optimal design is challenging due to the complexity of our design space: at each crossing of two wires, there is the binary choice of the layer ordering, i.e., which wire goes on top of the other. Furthermore, the rest shape of the elastic wires influences both the stability of the final object as well as the approximation quality of the target contours. To solve this problem efficiently, we propose to divide the optimization into three subproblems. First, we use a rigid body representation of the rods to determine the insertion order of wires and the layering at each crossing. This step also provides an indication as to whether achieving a stable structure is possible. Then, we switch to a deformable rod model and, for a given ordering, we optimize the shape of the wires in their final configuration to achieve contact configurations that are favorable for the structural stability, while trying to stay faithful to the desired target contours. Finally, we compute an optimal rest shape for each wire that balances the required force for assembly and stability. These shapes can be directly fabricated with a wire bending machine, and the physical prototype is ready to be assembled.

4 Method

In the following section, we will start by introducing basic notation and the formal problem definition, and then describe our method in detail.

4.1 Problem Setup and Notation

The user generates a representation of the 3D model based on a set $\mathcal{P}$ of P planar polylines $\Gamma^{(0)}, \ldots, \Gamma^{(P-1)}$. Each polyline $\Gamma^{(i)}$ consists of $n^{(i)}$ points $\mathbf{x}_0^{(i)}, \ldots, \mathbf{x}_{n^{(i)}-1}^{(i)}$ and $n^{(i)} - 1$ edges $\mathbf{e}_0^{(i)}, \ldots, \mathbf{e}_{n^{(i)}-2}^{(i)}$, with $\mathbf{e}_r^{(i)} = \mathbf{x}_{r+1}^{(i)} - \mathbf{x}_r^{(i)}$. As illustrated in Fig. 4, an intersection between two polylines, $\Gamma^{(i)}$ and $\Gamma^{(j)}$, generates a crossing c. This crossing involves points r and s in polylines i and j, respectively, and is characterized by its crossing point $\mathbf{x}_c = \frac{\mathbf{x}_r^{(i)} + \mathbf{x}_s^{(j)}}{2}$, and its incident edges $\mathbf{e}_{r-1}^{(i)}$, $\mathbf{e}_r^{(i)}$, $\mathbf{e}_{s-1}^{(j)}$ and $\mathbf{e}_s^{(j)}$. The distance between crossing points is given by the diameter of the wires. In practice, this distance is very small compared to the wire dimensions, and therefore we can safely neglect it in our stability computations.

At each crossing, two polylines are in contact. In order to obtain stable crossings (non-breaking contacts), so that the involved polylines do not come apart or slide away from each other, the contact force on one of them, produced by the other, must lie inside the corresponding *contact pyramid*. Each contact pyramid is characterized by the geometry of its crossing c and the frictional properties of the fabrication material. The geometric part is given by its apex, which is the crossing point $\mathbf{x}_c$, the bisector vectors $\mathbf{h}_c^{(i)}$ and $\mathbf{h}_c^{(j)}$, the contact angles $\alpha_c^{(i)}$ and $\alpha_c^{(j)}$, and the normals to the planes generating $\Gamma^{(i)}$ and $\Gamma^{(j)}$, $\mathbf{n}^{(i)}$ and $\mathbf{n}^{(j)}$. The bisector vector for polyline i (resp. j) is defined as $\mathbf{h}_c^{(i)} = (\hat{\mathbf{e}}_r^{(i)} - \hat{\mathbf{e}}_{r-1}^{(i)})/(\|\hat{\mathbf{e}}_r^{(i)} - \hat{\mathbf{e}}_{r-1}^{(i)}\|)$, with $\hat{\mathbf{e}}$ the corresponding normalized edge vector, while the contact angle for polyline i (resp. j) is $\alpha_c^{(i)} = \arctan\Big((\|\mathbf{e}_r^{(i)} \times (-\mathbf{e}_{r-1}^{(i)})\|)/(\mathbf{e}_r^{(i)} \cdot (-\mathbf{e}_{r-1}^{(i)}))\Big)$.

As illustrated in Fig. 4, taking into account only this purely geometric description and for now neglecting friction, the contact pyramid for polyline i (resp. j) at crossing c is defined by the set of vectors $\mathcal{B}_c^{(i)} = \left\{-\mathbf{b}_{c,r-1}^{(i)}, -\mathbf{b}_{c,r}^{(i)}, \mathbf{b}_{c,s-1}^{(j)}, \mathbf{b}_{c,s}^{(j)}\right\}$, which are determined by the in-plane edge normals $\mathbf{n}_{c,r-1}^{(i)}$, $\mathbf{n}_{c,r}^{(i)}$, $\mathbf{n}_{c,s-1}^{(j)}$ and $\mathbf{n}_{c,s}^{(j)}$, with $\mathbf{n}_{c,k}^{(l)} = (\mathbf{e}_k^{(l)} \times \mathbf{n}^{(l)})/(\|\mathbf{e}_k^{(l)} \times \mathbf{n}^{(l)}\|)$. Contact friction adds a tangential component to each basis vector, hence widening the pyramid by a factor $\alpha(\mu)$ that depends on the friction coefficient μ. This is shown in Fig. 4. We base our definition of static friction on the widely used Coulomb's law, which states that the friction force magnitude in the tangent plane at the contact interface f_t is related to the normal force magnitude f_n by $f_t \leq \mu f_n$ [Siciliano and Khatib 2008].

Stability Under External Loads For now, we assume that each polyline behaves rigidly once assembled into the final structure. Given a set of P rigid polylines $\mathcal{P}$ and the corresponding set of C crossings $\mathcal{C}$, we say the output structure is stable under some external load $\mathbf{l} \in \mathbb{R}^{6P}$ (representing per-polyline external force and torque) if there is a non-negative linear combination of contact pyramid vectors that cancels the external load, bringing the system into equilibrium. Since there may be a whole subspace of possible solutions to this problem, we look for the one that requires the smallest linear combination coefficients, therefore formulating the problem as a linear optimization problem:

$$\begin{aligned} \underset{\mathbf{a}}{\text{minimize}} \quad & \sum_i a_i \\ \text{subject to} \quad & \mathbf{M}\mathbf{a} = -\mathbf{l} \\ & \mathbf{a} \geq \mathbf{0}. \end{aligned} \tag{1}$$

The columns of $\mathbf{M}$, $\mathbf{m}_j \in \mathbb{R}^{6P}$, encode forces and torques produced by the contact pyramids' vectors. Each crossing introduces 4 contact pyramid vectors (columns), each with at most 12 non-zero

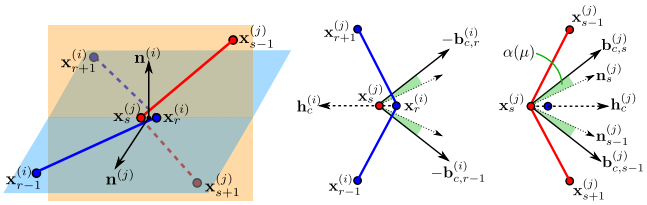

Figure 4: *Left: Schematic of the crossing for polylines $\Gamma^{(i)}$ and $\Gamma^{(j)}$. Right: 2D view of each polyline in their corresponding plane, indicating the in-plane normals and the friction-expanded contact pyramid vectors generated by polyline $\Gamma^{(j)}$.*

elements (two 6-tuples) representing force-torque vectors on the affected polylines. $\mathbf{a}$ represents the coefficients of the linear combination that balances the external load $\mathbf{l} \in \mathbb{R}^{6P}$. As we assume at this point that each polyline of the final assembled structure is a rigid body, $\mathbf{l}$ represents the per-polyline external loads, concatenating one 3D vector encoding external force and one 3D vector encoding the torque for each polyline. If at least one possible solution to this problem exists, we say the structure is stable under the given external load.

4.2 Layering and Assembly Order

In the first step of our wire sculpture design, we determine the layering of rods at contact points and the assembly order. At each crossing, two polylines intersect and their local ordering must be determined. In addition, stability is greatly influenced by the shape of the contact pyramids, which is determined by the local geometry of the polylines at the crossing points. As illustrated in Fig. 5, there are many different possible contact configurations, but only a few of them are stable in practice. Any stable structure using this limited set of contact configurations will result in a self-supporting assembly, not requiring additional connectors. Finally, we need to determine an assembly sequence that can be realized easily by a single person. In this section, we formulate a discrete problem that captures these goals and present an efficient branch-and-bound search strategy to solve it.

Optimal stable configuration A wire-based structure is stable when the contact forces produced by the contact pyramids can balance any possible external load up to a certain amount. Thus, to find a stable contact configuration, we search the space of all possible contact configurations. For each possible configuration, we evaluate stability for every possible external perturbation.

First, we will explain how to span the whole space of external perturbations.

Since the sum of all external loads must be zero (static equilibrium constraint), the space of all possible external loads $[\ldots \mathbf{f}_i^T \mathbf{t}_i^T \ldots]^T$ is a subspace of $\mathbb{R}^{6P}$, consisting of a force $\mathbf{f_i} = [f_{x,i}, f_{y,i}, f_{z,i}]$ and torque $\mathbf{t_i} = [t_{x,i}, t_{y,i}, t_{z,i}]$ per polyline i, such that $\sum_i \mathbf{f_i} = \mathbf{0}$ and $\sum_i \mathbf{t_i} = \mathbf{0}$. Its basis can be expressed by a $6 \cdot P \times 6 \cdot (P-1)$ matrix $\mathbf{L}$, where

$$\mathbf{L}_{ij} = \begin{cases} 1 & \text{if } i = j \\ -1 & \text{if } (i \bmod 6) = (j \bmod 6) \text{ and } i > 6 \cdot (P-1) \\ 0 & \text{otherwise.} \end{cases}$$

In order to be stable, a contact configuration must satisfy the stability condition in Eq. 1 for every external load basis vector. However, this requires that $\mathbf{a} \in \mathbb{R}^+$. Therefore, to span the complete space, negative external loads need to be explicitly included in the basis.

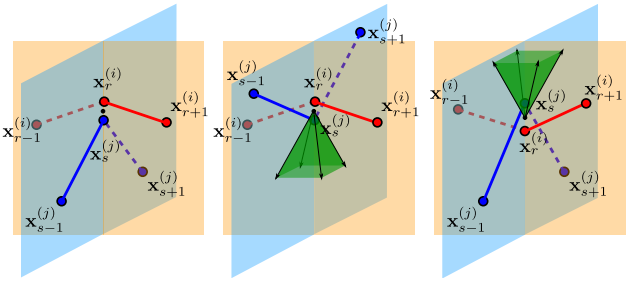

Figure 6: *Set of assigned possible contact pyramids and the corresponding modified geometries. Left: null contact without any geometric modifications (original crossing). Middle: Polyline i is modified to generate one of the possible contact configuration. Right: Polyline j is modified to generate the remaining contact configuration.*

In practice, instead of adding the negative of every basis vector in the positive half-space and solving $2 \cdot 6(P-1)$ linear programs, we simply extend $\mathbf{L}$ with a basis vector computed as the sum of the negatives of the basis vectors defined for the positive half-space.

To find an optimal stable configuration, we assign a set of n_p possible contact pyramids to each crossing. With C crossings, evaluating every combination requires ${n_p}^C$ tests. We limit the number of possible contact configurations to $n_p = 3$, namely the null (not effective) contact configuration, and the two contact pyramids representing opposite angular directions (Fig. 5). This, together with a highly effective branch-and-bound search strategy, allows us control the combinatorial explosion.

To obtain the desired contact pyramids, local geometric modifications near the crossing points in the involved polylines may be necessary. These modifications require changes ranging from a slight realignment of the bisector vectors through small changes in the adjacent points, to complete inversion of the bisector vectors on one or both polylines. These modifications are illustrated in Fig. 6 Our search strategy uses an optimality metric that measures these changes and explores the space of contact configurations aiming to minimize it.

Optimality metric There may be several possible stable configurations for a given contact set. As we assume that the elastic forces generated by an individual polyline are mainly in-plane of the polyline, we favor a crossing c where the contact pyramid is aligned with the intersection line of the two planes spanned by the two polylines (see Fig. 5). We define the optimal configuration as the one with the smallest maximum deviation from this quality metric, defined as:

$$e_c = \left(1 - \mathbf{h}_c^{(i)} \cdot \bar{\mathbf{h}}_c^{(i)}\right) + \left(1 - \mathbf{h}_c^{(j)} \cdot \bar{\mathbf{h}}_c^{(j)}\right), \qquad (2)$$

where $\mathbf{h}_c^{(i)}$ and $\bar{\mathbf{h}}_c^{(i)}$ (resp. j) are the bisector vectors for the possible contact pyramids and the input crossing configuration for polyline i (resp. j).

Assembly sequence We want the final structure to be easily assemblable, requiring no more than one person and no complex construction setup. This implies that the structure must be stable at every intermediate step in the assembly process. We represent the structure during the assembly process as a tree where the root contains as children all possible combinations of two polylines. The children of all other nodes span the combinations obtained by adding one of the remaining polylines. We explore this tree using a branch-and-bound search strategy. We use the following depth-first traversal strategy: at each node, we prioritize the children according

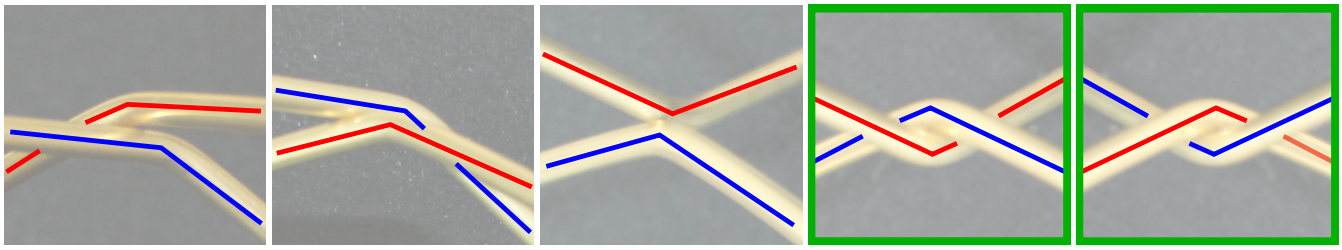

Figure 5: *Set of possible crossing configurations. The first three configurations (from left to right) produce narrow contact pyramids which are not stable in practice. The last two configurations are stable due to their wider contact pyramids and are the ones assigned as possible configurations on each crossing (together with the null contact).*

to our optimality metric. If the quality metric is better than the best complete structure found so far, we evaluate whether this configuration is stable and proceed along this path if positive. If we arrive at a leaf, we update the current best solution. In practice, this search strategy turned out to be highly effective, as it allows us to prune a large part of the combinatorial search space. Fig. 7 illustrates the construction and exploration of this tree.

Note that for the first level, with $P = 2$ polylines, the stability check will always fail when considering both forces and torques. The reason is that rotation around the intersection line given by the polylines' planes is unconstrained and therefore is evaluated as unstable. However, in practice, the rotation of one polyline is limited by the collision with the other involved polyline, and assembly is still possible. We include this consideration into the system by allowing translational-only stability at the root node, i.e., we evaluate stability for forces only, leaving rotational (torque) components out of the system. This situation can also happen with more polylines, but then assembly requires additional support, so we consider them unstable configurations.

While this optimization provides us the contact configuration and assembly sequence of a stable structure, the result is often not directly usable because it violates some fabrication constraints. Furthermore, in practice we observed that with some small local modifications, the stability of a figure can be greatly increased. Serving as an excellent initial guess, we use the result of contact configuration and assembly optimization to bootstrap our continuous target shape optimization, as explained in the following section.

4.3 Target Shape

At this point we have a stable contact configuration. Next, we have to adapt the target shape of the polylines (edges and angles) to fulfill fabrication and design constraints and improve the shape of the contact pyramids for increased stability. Later, as explained in Section 4.4, the fabrication-ready rest shapes will be computed based on the adapted polylines (where only angles change).

Target shape optimization could be solved by modifying the geometry locally, near the contact points. However, we must take into account fabrication and design constraints, which have a global effect. Hence, we formulate a global nonlinear constrained optimization problem for each polyline in the 2D domain:

$$\begin{aligned} \underset{\mathbf{p}}{\text{minimize}} \quad & E(\mathbf{p}, \bar{\mathbf{p}}) = E_d(\mathbf{p}, \bar{\mathbf{p}}) + w_r \cdot E_r(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{c}_f(\mathbf{p}) \geq 0 \\ & \mathbf{c}_c(\mathbf{p}) \geq 0, \end{aligned} \tag{3}$$

where E consists of a distance E_d and regularizer E_r energy term to be minimized, $\mathbf{c}_f$ are fabrication constraints, $\mathbf{c}_c$ are contact constraints, and $\mathbf{p}$ and $\bar{\mathbf{p}}$ are the points of the optimized and target polylines, respectively.

The goal of the polyline optimization is to generate polylines that comply with the diverse constraints while being as close as possible to the original target ones. We obtain $\bar{\mathbf{p}}$ by projecting the points $\bar{\mathbf{x}}$ of the user-specified contour $\bar{\Gamma}$ into a local coordinate frame that is aligned with the plane of the polyline. To initialize $\mathbf{p}$, we keep the original crossing points and resample the in-between subpolylines uniformly based on a user-defined point density.

Distance Energy To measure the distance between the source and optimized polylines in the 2D domain, we build a distance map, as in [Skouras et al. 2014], using implicit moving least squares [Öztireli et al. 2009]:

$$E_d(\mathbf{p}, \bar{\mathbf{p}}) = \sum_i w_i \frac{\sum_k \|\mathbf{n}_k(\mathbf{p}_i - \bar{\mathbf{p}}_k)\|^2 \phi_k(\mathbf{p}_i)}{\sum_k \phi_k(\mathbf{p}_i)}, \tag{4}$$

where $\phi_k(\mathbf{p}_i) = \max\left(0, \left(1 - \frac{\|\mathbf{p}_i - \bar{\mathbf{p}}_k\|_2^2}{h^2}\right)^4\right)$ are kernel functions with local support, vanishing beyond distance h, which is set to twice the distance of the average sampling distance, $\mathbf{n}_k$ are the normal vectors at the target polyline points, and $w_i = (\|p_i - p_{i-1}\| + \|p_{i+1} - p_i\|)$ is a weight proportional to the length of the incident line segments.

Regularization To promote a regular sampling and prevent abrupt changes in direction and noisy outputs with high point densities, we regularize the deformation based on the energy as follows:

$$E_r(\mathbf{u}) = \Delta \mathbf{u}_i = \sum_i \mathbf{u}_i - \frac{1}{2}(\mathbf{u}_{i-1} + \mathbf{u}_{i+1}), \tag{5}$$

where Δ is the uniform discrete Laplacian operator and $\mathbf{u}$ is the vector of displacements, measuring the displacement of each vertex w.r.t. its initial guess.

Fabrication constraints The fabrication constraints are determined by the underlying fabrication technology. Most wire bending machines work by sequential operations of either forwarding the wire or bending it. This limits the type of fabricatable curves to a discrete set of bending points with a required minimum distance between bending points. Furthermore, the maximum bending angle at each bending point is limited by the mechanics of the bending device. We model the distance, $c_{f,length} = \|\mathbf{p}_{i+1} - \mathbf{p}_i\| - l_{min} \geq 0$, and the maximum bending angle, $c_{f,angle} = \alpha_{max} - \alpha(\mathbf{p}) \geq 0$, as hard constraints. Note that the distance in the final configuration and rest configuration is the same, as we assume our wires to be inextensible. However, at this point, the final bending angles, which depend on the rest configuration, are not known. Consequently, if

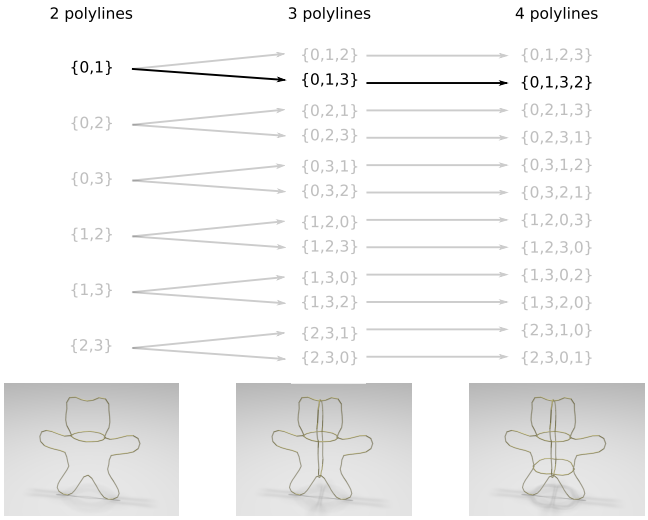

Figure 7: *Representation of the assembly tree for the teddy bear structure with $P = 4$ polylines. The search tree is initialized with all possible combinations of pairs of polylines. Child nodes are computed as all possible combinations of the remaining polylines for each parent node. The algorithm traverses this tree according to the optimality metric finding a stable assembly sequence.*

finally any angle constraint is violated, we update α_{max} conservatively and iterate starting from the shape optimization step. In practice none of our experiments required additional iterations.

Contact constraints The contact constraints are responsible for ensuring that the final structure produces the desired contact pyramids. For a given crossing c and involved polyline i, the desired contact pyramid is defined by the crossing point $\mathbf{p}_c^*$, the angle bisector $\mathbf{h}_c^{(i)*}$, and contact angle $\alpha_c^{(i)*}$. We fix the position of the crossing points, so that when mapping back to the 3D space, the polylines intersect at the crossing points, $\mathbf{p}_c^{(i)} - \mathbf{p}_c^* = \mathbf{0}$. The design choice of fixing the crossing points allows us to solve independent, per-polyline shape optimizations, which are computationally much cheaper than a global problem. Alternatively, in theory, one could relax this constraint and solve a global optimization problem involving the DOFs of all polylines. However, this would come at the cost of higher computational demands, which we found less favorable. Nevertheless, if the distance between crossings along one polyline does not allow for an intermediate non-fixed point, the shape cannot be changed locally. In this case, our system will try to find a solution with inactive crossings. If no solution is found, the model could be scaled up to increase the distance between crossing points and allow for unconstrained points in-between.

In addition, we formulate one constraint on each incident edge, so that its angle w.r.t. the desired bisector is smaller than the desired half contact angle: $\frac{\alpha_c^*}{2} - \alpha_{c+}^{(i)} \geq 0$ and $\frac{\alpha_c^*}{2} - \alpha_{c-}^{(i)} \geq 0$, with

$$\alpha_{c\pm}^{(i)} = \arctan\left(\frac{\|(\mathbf{p}_{r\pm1}^{(i)} - \mathbf{p}_r^{(i)}) \times \mathbf{h}_c^*\|}{(\mathbf{p}_{r\pm1}^{(i)} - \mathbf{p}_r^{(i)}) \cdot \mathbf{h}_c^*}\right) + \alpha(\mu),$$

where $r \pm 1$ represents the points in the polyline next $(+)$ and previous $(-)$ to the crossing point.

We solve this optimization problem for each polyline using the *Knitro* nonlinear constrained optimization library, obtaining close to interactive perfomance and allowing the user to comfortably iterate over a given design. An example of the target shape optimization is shown in Fig. 8.

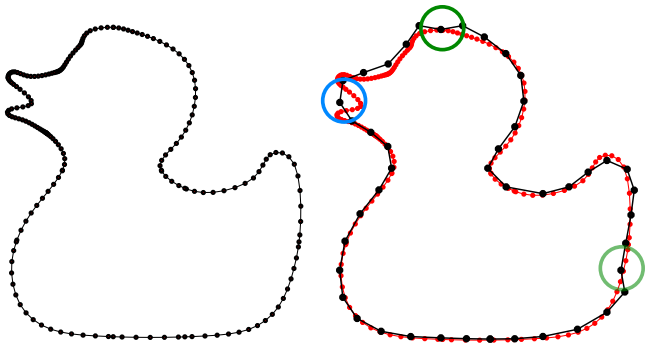

Figure 8: *Left: high-resolution polyline obtained from slicing the input triangle mesh. Right: overlay of polyline optimized with fabrication and contact constraints, and original polyline (red). In green, location of active contact contraints. In blue, maximum bending angle constraint. Minimum length constraints are active over the whole polyline.*

4.4 Internal Force and Polyline Rest Shape

Now that we have the target shape (a set of polylines generating a stable contact configuration together with the corresponding assembly sequence) we still have degrees of freedom left to determine the rest shape of the polylines. This allows us to determine the elastic deformation energy stored in the deformed polylines and thus optimize the internal force distribution and assemblability. Note that while the wires' endpoints should match in the target shape, they may not be coincident in the rest state.

Force optimization To maximize the stability of the final structure, we solve a nonlinear constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{a}}{\text{maximize}} \quad & G(\mathbf{a}) = G_{stb}(\mathbf{a}) + G_{asm}(\mathbf{a}) \\ \text{subject to} \quad & \mathbf{M}\mathbf{a} = \mathbf{0} \\ & |\mathbf{t}_v| \leq t_{max} \\ & \mathbf{a} \geq \mathbf{0}. \end{aligned} \tag{6}$$

The objective function $G(\mathbf{a})$ includes the structural stability and the assemblability objective functions, $G_{stb}(\mathbf{a})$ and $G_{asm}(\mathbf{a})$ respectively. To maximize the stability of the final structure, we maximize the normal component of the contact forces:

$$G_{stb}(\mathbf{a}) = \sum_{i \in \mathcal{P}} \sum_{k \in \mathcal{C}^{(i)}} |\mathbf{f}_k^{(i)} \cdot \mathbf{n}_k|, \tag{7}$$

where $\mathcal{C}^{(i)}$ is the set of crossings in polyline i. The contact force at crossing k is computed as

$$\mathbf{f}_k^{(i)} = \sum_{j \in \mathcal{B}^k} \mathbf{b}_j^{(i)} \cdot a_j$$

with $\mathcal{B}^k$ the set of contact pyramid vectors for crossing k and $\mathbf{b}_j^{(i)}$ the j-th basis vector, which acts on polyline i and is weighted by coefficient a_j of vector $\mathbf{a}$.

Physically, this objective function aims for a structure that will be stable under the largest possible external perturbations. However, the assembly also needs to fulfill several constraints. First, under no external load, it needs to be in static equilibrium, $\mathbf{M}\mathbf{a} = \mathbf{0}$.

Looking into the the internal contact forces and how they are produced, we realize that the deformation energy that each polyline

stores as it deforms from its rest configuration to the target configuration is what generates the internal contact forces. Since the target configuration is given by design, the only degrees of freedom available to modify the internal contact forces are the rest configurations of the polylines. However, the maximum applicable deformation is limited by the plastic limit of the fabrication materials.

We capture this limitation by imposing a constraint on the maximum bending torque on every point of every polyline, $|\mathbf{t}_v| \leq t_{max}$. We compute the total torque for point v in polyline i as the torque generated by the contact forces on each crossing on the subpolyline starting at point v and ending at the last point of the polyline. Formally, it is computed as

$$\mathbf{t}_v^{(i)} = \sum_{k \in \mathcal{C}^{(i)}, k > v} \left(\mathbf{x}_v^{(i)} - \mathbf{x}_k^{(i)} \right) \times \mathbf{f}_k^{(i)}.$$

Maximizing the largest external perturbation leads to very stable assemblies. However, it also tends to generate larger internal stress in the wires and increase the error near the wire cut points (i.e. wires do not close, leaving small gaps between start and end points). We allow the user to reduce the effective maximum bending torque, thus minimizing visual errors due to non-closing polylines while still obtaining stable assemblies.

For assemblability, we minimize the static equilibrium residual (i.e. force residual) at every intermediate assembly step s. Given a subset of polylines in assembly step s, the residual of the static equilibrium problem can be evaluated as $\|\mathbf{M}_s \mathbf{a}_s\|$. We integrate this term into the maximization problem as: .

$$G_{asm}(\mathbf{a}) = -\sum_s \|\mathbf{M}_s \mathbf{a}_s\|^2. \tag{8}$$

Finally, in closed polylines, contact forces are also greatly affected by the location of the polyline cut (start and end points). Our system provides an automatic suggestion for the cuts, placing them in the middle of the longest subpolyline. However, due to its large visual impact, we also allow the user to relocate it after polyline shape optimization.

It is worth mentioning that by solving Eq. 1 and Eq. 6 independently, we avoid a computationally expensive mixed discrete and continuous optimization problem. We focus on the discrete part when solving Eq. 1 and then on the continuous part for Eq. 6.

Rest shape computation Given the desired deformed configuration for each polyline in 2D, $\mathbf{p}$, and the contact forces (generated by the rest of the polylines) acting on it, $\mathbf{f}$, our goal is to compute the rest configuration that cancels those forces.

To solve the inverse problem of finding the rest configuration, we aim for torque equilibrium at every vertex v of the polyline i. We model each polyline as an inextensible elastic rod [Bergou et al. 2008] in 2D, which allows us to ignore the twist energy component, and formulate the elastic bending energy in terms of angles in the undeformed and deformed configurations, $\bar{\alpha}_v$ and α_v:

$$E_{bend} = \sum_v \frac{4}{\bar{l}_v} k_b \left(\tan\left(\frac{\alpha_v}{2}\right) - \tan\left(\frac{\bar{\alpha}_v}{2}\right) \right)^2, \tag{9}$$

where $\bar{l}_v = \frac{1}{2}\left(|\mathbf{e}_{v-1}| + |\mathbf{e}_v|\right)$ is the integration domain for the energy, $\alpha_v = \arctan\left(\frac{|\mathbf{e}_{v-1} \times \mathbf{e}_v|}{\mathbf{e}_{v-1} \cdot \mathbf{e}_v}\right)$ is the angle at every point v, and k_b is the bending stiffness.

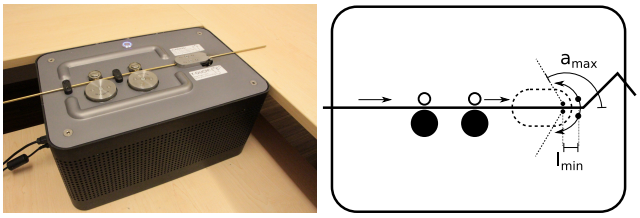

Figure 9: *DiWire Bending machine by PensaLabs and the schematic with the fabrication constraints.*

The (scalar) elastic bending torque at vertex v can then be computed as the negative gradient of the energy w.r.t. angle α_v:

$$t_v^{int} = -\frac{4}{\bar{l}_v} k_b \left(\tan\left(\frac{\alpha_v}{2}\right) - \tan\left(\frac{\bar{\alpha}_v}{2}\right) \right) \left(1 + \tan^2\left(\frac{\alpha_v}{2}\right)\right). \tag{10}$$

Given the elastic bending torque, we can formulate the torque equilibrium equation as follows:

$$t_v^{(i)} = \sum_{k \in \mathcal{C}^{(i)}, k > v} \left(\mathbf{p}_v^{(i)} - \mathbf{p}_k^{(i)} \right) \times \mathbf{g}_k^{(i)}, \tag{11}$$

where $\mathbf{g}_k^{(i)} = \mathbf{f}_k^{(i)} - \left(\mathbf{f}_k^{(i)} \cdot \mathbf{n}^{(i)}\right) \mathbf{n}^{(i)}$ is the projection of the force in 3D onto the 2D plane that defines polyline i.

The rest angle can then be computed as

$$\bar{\alpha}_v = 2 \cdot \arctan\left(\tan\left(\frac{\alpha_v}{2}\right) - \frac{\sum t_v^{ext}}{\frac{4}{\bar{l}_v} k_b \left(1 + \tan^2\left(\frac{\alpha_v}{2}\right)\right)} \right), \tag{12}$$

which fully defines the rest shape of the polyline.

5 Fabrication and Assembly

5.1 Wire Bending

To obtain the rest configurations for each individual polyline, we use Pensalabs' DiWire bending machine (see Fig. 9). This device consists of a wire feeder that pushes the input straight wire forward, a fixed set of pins, and a rotating bending head.

As input, the user provides a polyline description in 2D as a vector graphics file (SVG or DXF). During the fabrication process, the feeder either pushes the wire forward into the bending head, or the bending head rotates at a given angle, thereby bending the wire. In practice, we found self collisions of the wire a negligible problem, as the wire is elastic and is able to slightly slide out of plane, resulting in only minor fabrication errors. However, careful calibration of the viscoelastic behavior is essential due to the spring-back effect. We use a data-driven process. We feed the machine with known input angles and measure the output angles. Then, we use linear regression and invert this function to map the desired final angles to the compensated inputs.

The mechanics of the device impose constraints on the maximum bending angle and the minimum length between bending points. Theoretically, the maximum bending angle is limited by the maximum angle reached by the bending head. In practice, however, due to the *spring-back* effect of the wire, this maximum angle is smaller and determined empirically during the calibration process. Due to the springback effect and limitations in the measurement accuracy

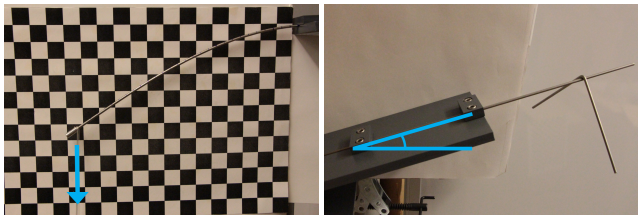

Figure 10: *Capture setups using steel wires. Left: capture setup for the bending stiffness and plasticity limit. Right: capture setup for the friction coefficient. The arrow and angle shown in blue illustrate the applied force and tilt, respectively.*

during calibration, there is also a minimum bending angle that can be accurately achieved, which in our case is around 1 degree.

The minimum length between bending points is constant, $l_{min} = 1.5$ cm, and given by the distance between the rotating head and the fixed pins. As seen in Fig. 9, this is the minimum distance required between bending points that guarantees that the wire will be straight at the beginning of the bending process.

5.2 Parameter Estimation

There are several material parameters that influence the rest configurations obtained from the design system: bending stiffness, bending torque for which the model enters into the plastic regime, and the friction coefficient. To obtain accurate results, we estimate these parameters from real wires.

We estimate these parameters using two very lightweight capture setups, as illustrated in Fig. 10. To estimate the bending stiffness and maximum torque, we fix a rod to a horizontal surface, leaving 30 cm off the support surface, and record the deformation for a set of different loads. We use loads ranging from 0 grams to 1000 grams, with 100 grams increments, and load and unload the wire for every step. Any change in the unloaded configuration indicates that the plastic limit has been reached. We set the allowed maximum bending torque to the torque produced by the maximum load that showed no deformation w.r.t. the initial configuration.

To estimate the bending stiffness, we consider plasticity-free deformations and compute the torque produced by the external load at a set of uniformly sampled points along the wire. Then, we minimize the difference w.r.t. the torques produced by the elastic bending model described in Section 4.4. Formally, we solve a least-squares minimization problem:

$$\underset{k_b}{\text{minimize}} \sum_{\mathcal{F}} \sum_{v} \left(t_v^{int} - t_v^{ext}\right)^2, \tag{13}$$

where $\mathcal{F}$ is the set of captured load frames and v traverses the set of evaluation nodes.

After estimation, we evaluated the fitness of the estimated parameters and obtained a position RMS error of 5.2 mm for brass and 1.9 mm for steel.

To estimate the friction coefficient, we attach a rod to an inclinable surface in its $\alpha = 0$ degree horizontal configuration and place a V-shaped rod on top of it. Then, we slowly increase the inclination of the surface until the V-shaped rod starts sliding and record the obtained angle α. The friction coefficient can then be computed as $\mu = \tan\alpha$. Table 1 shows the values obtained for brass and steel wires.

	k_b $(N \cdot m^2)$	Max. Torque $(N \cdot m)$	μ
Brass	0.42	1.30	0.23
Steel	0.73	1.77	0.32

Table 1: *Estimated parameters for brass and steel wires with* 3 *mm diameter.*

6 Results

We have used our system to design several wire sculptures. For validation, we created physical prototypes of all of our models, with a subset shown in Fig. 12. The models are made out of brass and steel wires with a diameter of 3mm and fabricated with a consumer-level computer-controlled wire bending machine (Pensalabs DiWire bending machine, Fig. 9). In the following, we will validate our design choices and discuss our results in more detail.

Stability To validate our approach, we have built and tested several physical prototypes. To systematically test the stability, we performed several drop tests with the model "Sphere". We created three different versions, with torque values of 30%, 60% and 100% of the real maximum torque, which survived drops up to 30 cm, 80 cm and 100 cm, respectively. In general, models made out of steel are more stable than models made out of brass, as steel can store a much higher elastic energy resulting in higher internal forces. All models are sufficiently stable for being handled without the need of extra care, and can for example be lifted or touched at common acceleration rates. Note that although a given set of initial contours might be stable, the resistance to external loads might be close to non-existent. For all models shown in the paper, the initial contours even with an optimal layering would not be able to resist gravity. Optimizing for improved stability is therefore essential for achieving a functional design in practice.

Performance of Optimization To evaluate the performance of our optimization, we start from a given set of contours, and measure the required computational time for finding the layering and assembly order, rest shape optimization, and target shape optimization. All computations were done on a standard desktop computer with 3.80 GHz and a single thread. We use GLPK and KNitro to solve linear programs and nonlinear constrained optimization, respectively. On average, computing a figure takes between a few seconds to a few minutes. Detailed timings and statistics for some of our models can be found in Table 2. Usually, we run shape target optimization and rest shape optimization only once. However, in theory they are coupled because the maximum bending angle as fabrication constraint is required in the target shape optimization but defined with respect to the rest shape. In rare cases where this is an issue, we iterate between target and rest shape optimization, and adjust the maximum angle accordingly. While not fully interactive, our prototype implementation is sufficiently fast to provide users the option to compare various designs, allowing them to make an informed decision on aesthetics and stability of the resulting figure. Besides the number of wires and crossing points, we found that the scale of the model influences the required computation time significantly, as the minimum length between bending points is fixed due to fabrication constraints. Table 2 shows that for the airplane model the time spent solving the shape optimization is larger than for the other models. The reason is that two of the polylines composing this model are really long and therefore, for a given point density, the total number of degrees of freedom (size of the optimization problem) is much bigger.

Fabrication Process Our fabrication process is low-cost and fast. On average, forwarding the wire to the next bending point

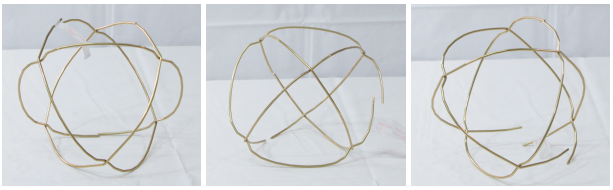

Figure 11: *Drop test. Three different sphere models, designed with (from left to right)* 30%*,* 60% *and* 100% *of the real maximum torque are stable when being dropped from increasing heights, breaking at* 30 *cm,* 80 *cm and* 100 *cm heights, respectively.*

	P	C	l_{max} (cm)	Stab. (s)	S.Opt. (s)	R.Conf. (s)
Duck	5	14	80.1	0.19	2.21	7.54
Teddy	4	10	125.4	0.09	7.99	0.02
Airplane	6	18	209.5	0.60	50.91	5.86
Car	6	22	117.5	1.83	10.04	16.12
Chair	8	38	133.1	22.22	11.47	95.76

Table 2: *Statistics and timings for our models.*

and performing the actual bend takes approximately three seconds, resulting in a total fabrication and assembly time between less than 12 minutes ("Teddy") for our simplest model and less than 25 minutes ("Chair") for our most complex object. Accurately calibrating the machine is extremely important. Nevertheless, each forward and bending motion introduces a small error due to mechanical inaccuracies. This error accumulates and is noticeable at locations where a wire loop should meet. While we did not do any manual correction, this could be fixed by slight manual adjustments. Alternatively, to preserve the aesthetically important smooth flow lines of the car model, we used simple connectors to close the loops. Note that also in this case we did not use any connectors at contact points between wires, and the model is fully self-supporting due to the optimal arrangement of the wires.

Assembly Although we cannot guarantee a collision-free assembly path for all wires, a problem which we guess to be computationally intractable due to the similarity to the computational hardness of motion planning, we did not find this an issue in our experiments due to the elasticity of the wires. Out-of-plane components of the contact forces can lead to more difficult assembly processes. In practice, however, we did not find any problematic cases. Numerically, we obtained an average per-assembly out-of-plane component to maximum contact force ratio of 5%, with a maximum of 6.3% for the duck model. Our search strategy for assembly sequences is limited to a search space involving the addition of single polylines to an existing structure. This assumption could be too restrictive in some cases. Consider a torus composed of 4 circular wires obtained from slicing planes perpendicular to the revolution axis (one plane splitting the torus in half, two planes touching the top and bottom of the torus) and several circular wires from radial planes. There exists a stable final configuration but our method will not be able to find a valid assembly sequence because every stable intermediate assembly requires more than 3 polylines.

7 Conclusion

We have presented a computational design system for creating stable, self-supporting rod structures assembled from in-plane bent wires. Our model incorporates the relevant physical effects of the elastic wires, frictional contact, and the fabrication constraints of the underlying manufacturing technology. Our effective method for sequentially optimizing the discrete contact order, the continuous target shape, and the internal contact force magnitude allowed us to design compelling sculptures which resemble the desired shape while at the same time complying with a stable equilibrium. As demonstrated by our results, we successfully fabricated all shown models and evaluated the stability by performing drop tests with several physical models.

Limitations and Future Work By design, individual wires are expected to be in a plane. We also assume that elastic deformations are restricted to stay in-plane, which turned out to be a reasonable approximation for all the examples we tested in practice. There is also a minimum size limitation due to the fabrication technology: reproducing very thin features requires small contours with high curvature, which may violate fabrication constraints. In these cases, the model may need to be scaled to a large size. Similarly, the distance between two crossings along one polyline must be large enough to accommodate at least one non-crossing point in-between, so that the system has enough degrees of freedom to find a solution for the different optimization subproblems while fulfilling the fabrication constraints. For future work, we plan to extend our model to arbitrary 3D wire paths. Although computer-controlled 3D wire bending machines are currently expensive and only available in professional industry settings, we expect that 3D paths enable highly interesting additional expressive freedom for artists and designers. Another limitation is that small fabrication inaccuracies of our wire bending machine can accumulate for long wires. Although more of an engineering problem, this could easily be solved by using for example a vision system and active feedback during the fabrication. Currently, the input contours are computed by intersecting planes with a surface model. While this provides a simple and effective way to interface the system, in the future more expressive input modalities could be explored, building on research in non-photorealistic rendering, sketching, and haptic interfaces. Currently, we use an elastic bending model, which is able to accurately describe small-scale deformations. For more accurate large-scale deformations, one could investigate more sophisticated nonlinear constitutive models. A key aesthetic feature of our sculptures is the minimalistic representation of shape, reduced to a few lines. However, in theory they could also serve as a frame for supporting and giving shape to a cover. In general, we think an exciting direction for future work would be to explore further applications of our approach by integrating design capabilities for higher-level functionality, such as furniture.

Acknowledgements

We would like to thank everyone who contributed to this paper, especially Ruslan Guseinov, Thomas Auzinger and Ran Zhang for their help with the final demos and renders, as well as all proof-readers and anonymous reviewers. We thank *Jakobi* for allowing us to use the image of his dog wire sculpture. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645599.

References

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. In *ACM transactions on graphics (TOG)*, vol. 27, ACM, 63.

BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics (TOG) 25*, 3, 1180–1187.

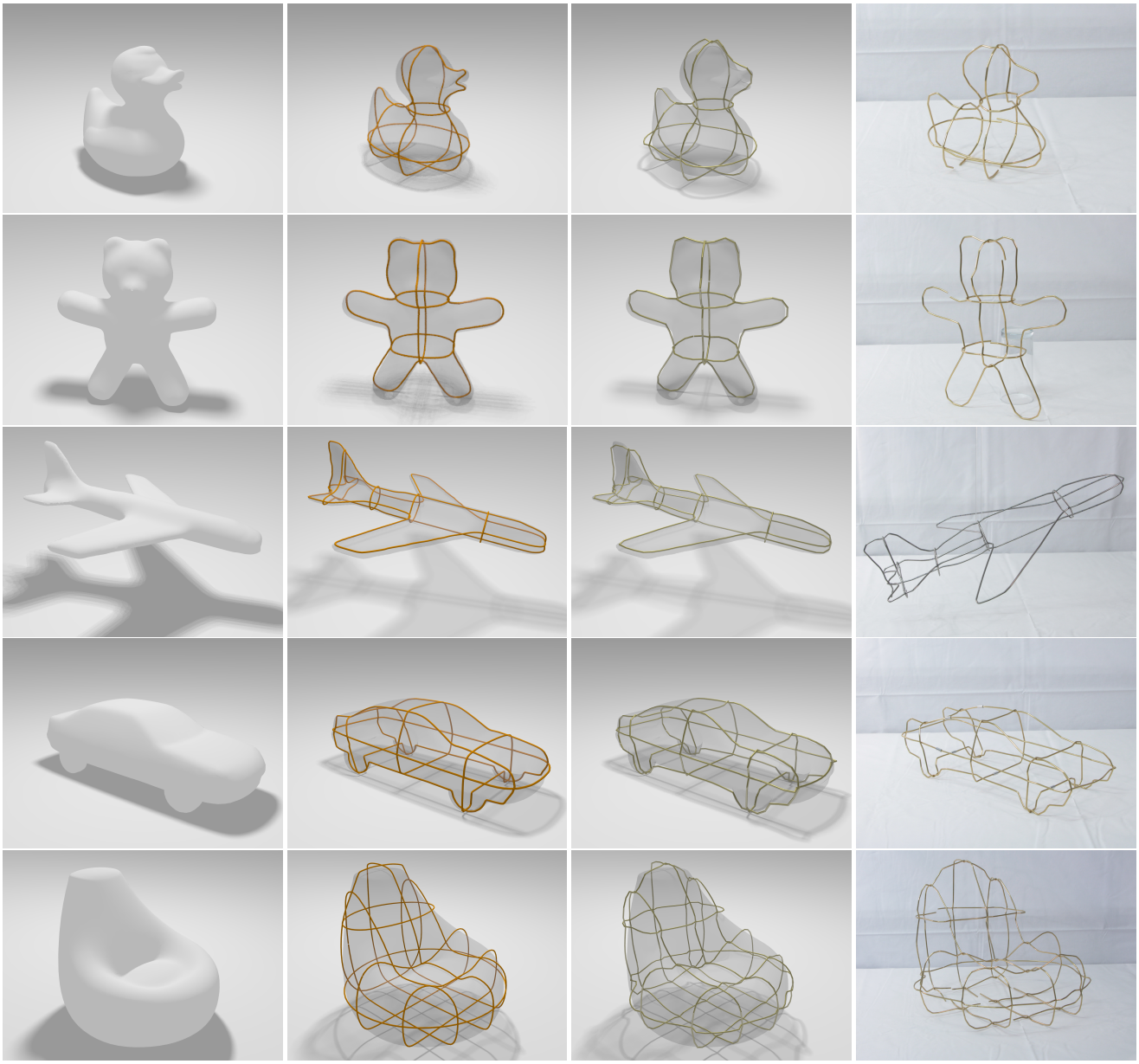

Figure 12: *Fabricated structures at different stages of the pipeline. From left to right: input model, sliced contours, optimized rod-structure, fabricated model.*

CIGNONI, P., PIETRONI, N., MALOMO, L., AND SCOPIGNO, R. 2014. Field-aligned mesh joinery. *ACM Transactions on Graphics (TOG) 33*, 1, 11.

DE GOES, F., ALLIEZ, P., OWHADI, H., AND DESBRUN, M. 2013. On the equilibrium of simplicial masonry structures. *ACM Transactions on Graphics (TOG) 32*, 4, 93.

DEMAINE, E. D., AND ORourke, J. 2007. *Geometric folding algorithms*. Cambridge university press Cambridge.

DEROUET-JOURDAN, A., BERTAILS-DESCOUBES, F., DAVIET, G., AND THOLLOT, J. 2013. Inverse dynamic hair modeling with frictional contact. *ACM Transactions on Graphics (TOG) 32*, 6, 159.

DEUSS, M., PANOZZO, D., WHITING, E., LIU, Y., BLOCK, P., SORKINE-HORNUNG, O., AND PAULY, M. 2014. Assembling self-supporting structures. *ACM Transactions on Graphics (TOG) 33*, 6, 214.

FRICK, U., VAN MELE, T., AND BLOCK, P. 2015. Decomposing three-dimensional shapes into self-supporting, discrete-element assemblies. In *Modelling Behaviour*. Springer, 187–201.

FU, C.-W., SONG, P., YAN, X., YANG, L. W., JAYARAMAN, P. K., AND COHEN-OR, D. 2015. Computational interlocking furniture assembly. *ACM Transactions on Graphics (TOG) 34*, 4, 91.

GARG, A., SAGEMAN-FURNAS, A. O., DENG, B., YUE, Y., GRINSPUN, E., PAULY, M., AND WARDETZKY, M. 2014. Wire mesh design. *ACM Transactions on Graphics (TOG) 33*, 4, 66.

GRÉGOIRE, M., AND SCHÖMER, E. 2007. Interactive simulation of one-dimensional flexible parts. *Computer-Aided Design 39*, 8, 694–707.

HADAP, S. 2006. Oriented strands: dynamics of stiff multi-body system. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, 91–100.

HILDEBRAND, K., BICKEL, B., AND ALEXA, M. 2012. crdbrd: Shape fabrication by sliding planar slices. In *Computer Graphics Forum*, vol. 31, Wiley Online Library, 583–592.

IARUSSI, E., LI, W., AND BOUSSEAU, A. 2015. Wrapit: computer-assisted crafting of wire wrapped jewelry. *ACM Transactions on Graphics (TOG) 34*, 6, 221.

IBEN, H., MEYER, M., PETROVIC, L., SOARES, O., ANDERSON, J., AND WITKIN, A. 2013. Artistic simulation of curly hair. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 63–71.

KILIAN, M., FLÖRY, S., CHEN, Z., MITRA, N. J., SHEFFER, A., AND POTTMANN, H. 2008. Curved folding. In *ACM Transactions on Graphics (TOG)*, vol. 27, ACM, 75.

LAU, C., SCHWARTZBURG, Y., SHAJI, A., SADEGHIPOOR, Z., AND SÜSSTRUNK, S. 2014. Creating personalized jigsaw puzzles. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering*, ACM, 31–39.

LIU, Y., PAN, H., SNYDER, J., WANG, W., AND GUO, B. 2013. Computing self-supporting surfaces by regular triangulation. *ACM Transactions on Graphics (TOG) 32*, 4, 92.

MASSARWI, F., GOTSMAN, C., AND ELBER, G. 2007. Papercraft models using generalized cylinders. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, IEEE, 148–157.

MITANI, J., AND SUZUKI, H. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 259–263.

ÖZTIRELI, A. C., GUENNEBAUD, G., AND GROSS, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*, vol. 28, Wiley Online Library, 493–501.

PAI, D. K. 2002. Strands: Interactive simulation of thin solids using cosserat models. In *Computer Graphics Forum*, vol. 21, Wiley Online Library, 347–352.

PANOZZO, D., BLOCK, P., AND SORKINE-HORNUNG, O. 2013. Designing unreinforced masonry models. *ACM Transactions on Graphics (TOG) 32*, 4, 91.

PÉREZ, J., THOMASZEWSKI, B., COROS, S., BICKEL, B., CANABAL, J. A., SUMNER, R., AND OTADUY, M. A. 2015. Design and fabrication of flexible rod meshes. *ACM Transactions on Graphics (TOG) 34*, 4, 138.

SAGEMAN-FURNAS, A. O., UMETANI, N., AND SCHMIDT, R. 2015. Meltables: fabrication of complex 3d curves by melting. In *SIGGRAPH Asia 2015 Technical Briefs*, ACM, 14.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. In *ACM Transactions on Graphics (TOG)*, vol. 27, ACM, 64.

SICILIANO, B., AND KHATIB, O. 2008. *Springer handbook of robotics*. Springer Science & Business Media.

SKOURAS, M., THOMASZEWSKI, B., BICKEL, B., AND GROSS, M. 2012. Computational design of rubber balloons. In *Computer Graphics Forum*, vol. 31, Wiley Online Library, 835–844.

SKOURAS, M., THOMASZEWSKI, B., KAUFMANN, P., GARG, A., BICKEL, B., GRINSPUN, E., AND GROSS, M. 2014. Designing inflatable structures. *ACM Transactions on Graphics (TOG) 33*, 4, 63.

SKOURAS, M., COROS, S., GRINSPUN, E., AND THOMASZEWSKI, B. 2015. Interactive surface design with interlocking elements. *ACM Transactions on Graphics (TOG) 34*, 6, 224.

SONG, P., FU, C.-W., AND COHEN-OR, D. 2012. Recursive interlocking puzzles. *ACM Transactions on Graphics (TOG) 31*, 6, 128.

TWIGG, C. D., AND KAČIĆ-ALESIĆ, Z. 2011. Optimization for sag-free simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 225–236.

VOUGA, E., HÖBINGER, M., WALLNER, J., AND POTTMANN, H. 2012. Design of self-supporting surfaces. *ACM Transactions on Graphics (TOG) 31*, 4, 87.

WEYRICH, T., DENG, J., BARNES, C., RUSINKIEWICZ, S., AND FINKELSTEIN, A. 2007. Digital bas-relief from 3d scenes. In *ACM Transactions on Graphics (TOG)*, vol. 26, ACM, 32.

WHITING, E., OCHSENDORF, J., AND DURAND, F. 2009. Procedural modeling of structurally-sound masonry buildings. *ACM Transactions on Graphics (TOG) 28*, 5, 112.

WHITING, E., SHIN, H., WANG, R., OCHSENDORF, J., AND DURAND, F. 2012. Structural optimization of 3d masonry buildings. *ACM Transactions on Graphics (TOG) 31*, 6, 159.